

# FPGA BASIERTE, KONFIGURIERBARE OFDM SENDER-PLATTFORM FÜR DIE POSITIONSBESTIMMUNG MITTELS TDOA

E. Staudinger, C. Klein, H. Fröhlich, J. A. Jungreithmeier, W. Dichler, A. Klopff, H. G. Brachtendorf, A. Dammann<sup>1</sup>

FH OÖ Studienbetriebs GmbH, 4232 Hagenberg, Österreich

<sup>1</sup>DLR Institut für Kommunikation und Navigation, 82234 Weßling, Deutschland

## Zusammenfassung

Für die Evaluierung von Algorithmen zur Positionsbestimmung nach dem *Time Difference of Arrival* – Verfahren wird eine frei konfigurierbare OFDM Sender-Plattform benötigt. Da die Berechnung in Echtzeit erfolgen soll, ist eine Implementierung in rekonfigurierbarer Hardware (FPGA) erforderlich. Das Manuskript gibt sowohl einen Überblick über die Hintergründe und die Architektur des Sendesystems, als auch einen tieferen Einblick in verschiedene Lösungsdetails.

## 1. EINFÜHRUNG

Positionsbestimmungen basierend auf dem *Global Navigation Satellite System* (GNSS) wie z.B. GPS oder GALILEO liefern im freien Feld und im Großstadtbereich ausreichend exakte Ergebnisse für die meisten Positionierungsanwendungen. In diesen Szenarien sind die Straßenschluchten vergleichsweise breit und die Anzahl der sichtbaren Satelliten ist ausreichend hoch.

Im Innenstadtbereich mit engen Straßenschluchten oder im Indoor-Bereich sind nur sehr wenige bis gar keine Satelliten sichtbar. Eine Positionierung in diesen Szenarien ist mit dem GNSS kaum möglich und muss mit zusätzlichen Systemen ergänzt werden. Hierzu bieten sich aktuelle und im Aufbau befindliche Funkssysteme auf OFDM-Basis an. Des Weiteren ist die Zeitdauer einer ersten zuverlässigen Positionsschätzung bei GNSS-basierten Systemen vergleichsweise hoch (im Minutenbereich) [1; 2].

Standards wie 3GPP-LTE, 802.11x, DVB-T, DVB-H, WiMAX etc. sind OFDM basiert und bieten gegenwärtig eine hohe Nutzsignal-Bandbreite von bis zu 20 MHz an. Zur Identifizierung unterschiedlicher Sendestationen werden zudem meist zusätzlich spezifische Synchronisationssequenzen mit entsprechenden Korrelationseigenschaften abgestrahlt. Diese können für eine Positionsschätzung und zur Schätzung von Mehrwegekanälen herangezogen werden [3]. Zur Positionsschätzung existieren bereits entsprechende Verfahren wie z.B. *Time Difference of Arrival* (TDoA), *Angle of Arrival* (AoA) und *Time of Arrival* (ToA) [4]. Im weiteren Verlauf wird nur TDoA näher betrachtet. Bei diesem Verfahren müssen die beteiligten Sendestationen zeitlich synchronisiert sein, der mobile Empfänger jedoch nicht mit den Sendern. Dies reduziert die Komplexität im mobilen Empfangsgerät. Die gemessenen Zeitdifferenzen führen zu Hyperbeln in deren Foki sich die Mobilstation befindet [5], siehe Bild 1.

Bei einem frei konfigurierbaren Sendesystem auf OFDM-Basis sollten folgende Parameter variabel gestaltet sein:

- Anzahl der Subträger, Länge des Cyclic-Prefix,
- Länge eines OFDM-Rahmens,

- Verwendung einer speziellen Präambel mit gesondertem Cyclic-Prefix,
- synchrone Abstrahlung der Daten von 1 bis 4 Sendern,
- frei wählbare Symbol-Zuordnung (BPSK, QPSK, 16-QAM, 64-QAM, 128-QAM, etc.) und
- einfache Konfiguration der Sender über einen PC mit z.B. Ethernet-Schnittstelle.

Mit einem solchen System lassen sich somit nahezu alle gegenwärtigen Funkstandards ohne spezielles Zeitzugriffsverfahren nachbilden. Des Weiteren bietet es eine Basis für die Realisierung eines entsprechenden Empfängers.

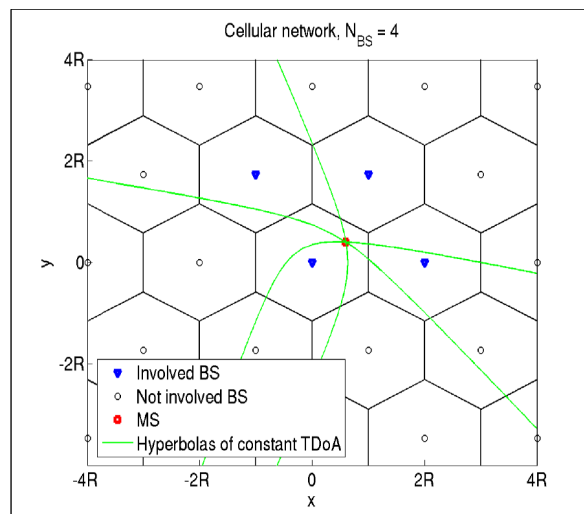


Bild 1: TDoA Positionierung in einem zellenbasierten Netzwerk mit 4 beteiligten Basisstationen BS und einer Mobilstation MS [6].

Das Manuskript ist in folgende Kapitel unterteilt: Das zweite Kapitel befasst sich mit dem Systemkonzept, das dritte mit den Einzelheiten des FPGA-Designs und das vierte mit der Anpassung an die DAC-Kanäle. Es folgen

Details zur Synchronisation, der PC-Software und der Firmware. Den Abschluss bildet das achte Kapitel mit den Ergebnissen der Implementierung.

## 2. SYSTEMKONZEPT

Als Plattform für die Sender wurde das Stratix III DSP Development Kit von Altera gewählt. Dieses enthält bereits eine Zusatzplatine mit Analog/Digital- als auch Digital/Analog-Wandlern (250 MSamples/s) wobei der DAC aus einem Chip mit zwei unabhängigen Kanälen besteht. Mit diesen zwei Kanälen können entweder ein komplexes Basisbandsignal oder bereits zwei digital hochgemischte Zwischenfrequenzsignale (ZF) ausgegeben werden. Zur Reduktion der Hardware-Komponenten wird die Zwischenfrequenzlösung gewählt. Für die Realisierung der geforderten 4 Sender werden somit zwei FPGA-Boards mit je einem DAC-Chip benötigt. Die ZF sollte zwischen 60 MHz und 70 MHz liegen, was in der Funktechnik einen gebräuchlichen Wert darstellt. Die digitalen, reellwertigen ZF-Signale werden digital/analog gewandelt und auf z.B. 5.2 GHz hochgemischt (MX). Das Hochfrequenzsignal wird anschließend verstärkt (PA) und den Antennen zugeführt, siehe Bild 2.

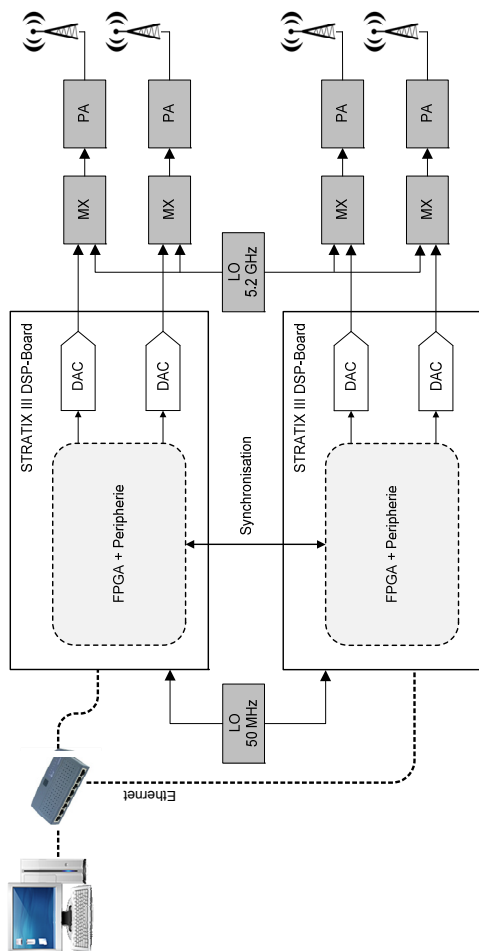


Bild 2: Übersicht des Gesamtdesigns, Systemkonzept.

Ein Hauptkriterium beim Systementwurf war die Konfiguration sämtlicher OFDM-spezifischer Parameter zur Laufzeit des Systems. Die in Kap. 1 geforderten Parameter werden folgendermaßen umgesetzt:

- Subträger-Anzahl von 64 bis 8192 in 2er-Potenzen,
- Cyclic-Prefix-Länge von einem bis zur Anzahl der Subträger,
- wählbare OFDM-Rahmenlänge von 32 oder 64 OFDM-Symbolen,
- gesonderte Behandlung der ersten OFDM-Symbole (Präambel) mit gesondertem Cyclic-Prefix,
- frei wählbare Systembandbreite durch externe Taktquelle (hier fixiert auf 20 MHz Bandbreite) und
- frei wählbare Signal-Konstellationen.

Der letzte Punkt bezüglich der Konstellationen wird folgendermaßen generisch realisiert: Für jeden Subträger vor der IFFT werden 8 Bits für den Real- als auch 8 Bits für den Imaginärteil verwendet. Damit lässt sich, unter Berücksichtigung dieser Quantisierung, nahezu jedes Konstellationsdiagramm realisieren. Wurde ein OFDM-Rahmen vollständig gesendet, wird dieser permanent wiederholt, bis das System gestoppt, oder neu konfiguriert wird. Die IFFT wird am FPGA in Echtzeit gerechnet, da nur die Daten im Frequenzbereich vom PC zum FPGA übertragen werden. Dies setzt eine Realisierung der IFFT in Hardware (Logikelementen) voraus.

Die Übertragung vom PC zu den beiden FPGA-Boards könnte, aus Gründen der Einfachheit, über eine konventionelle serielle Schnittstelle (RS232) erfolgen. Die Datenmenge zur Konfiguration, sowie die geringe Leitungslänge bei hohen Datenraten der RS232 limitieren jedoch deren Einsatz. Als Schnittstelle wird Ethernet mit 100 Mbit/s gewählt. Die Vorteile hierbei liegen in der höheren Datenrate, der größeren Leitungslänge und der meist vorhandenen Gebäudeverkabelung. Der PC lässt sich somit an beliebiger Stelle, getrennt von den Sendern, aufstellen. Das Stratix III DSP Board verfügt bereits über einen Ethernet-PHY, welcher die physikalische Schicht darstellt. Zusätzlich wird ein Ethernet-MAC zur Verarbeitung der Sicherungsschicht (Layer 2) benötigt. Hierfür wird auf den Triple-Speed Ethernet MegaCore von Altera zurückgegriffen. Die verbleibenden Schichten sind in Software implementiert, wofür ein Prozessor im FPGA benötigt wird. Da das FPGA über keinen eingebauten Prozessor verfügt, wird eine Softcore-CPU eingesetzt. Diese CPU besteht aus Logikelementen des FPGAs und kann auf verschiedenen FPGAs mit unterschiedlichen Konfigurationen realisiert werden. Dabei wird der NIOS II Softcore-Prozessor von Altera als CPU verwendet.

Ein weiteres wichtiges Entwurfskriterium ist die zeitliche Synchronität aller 4 Sender. Die zwei Sender je FPGA sind per Definition zeitsynchron, da sie sich stets in derselben Taktdomäne befinden. Zur Vereinfachung der Anforderungen an die Synchronität werden beiden Boards mit ein und derselben Taktquelle gespeist. Somit entfällt die Frequenzsynchronisation und es müssen nur noch Laufzeiten berücksichtigt werden, siehe Kap. 5.

## 3. SOC – DESIGN

Ein in einem FPGA realisiertes und aus mehreren Komponenten bestehendes System wird auch als *System*

on Chip (SOC) bezeichnet. Dieses SOC lässt sich grob in folgende Komponenten aufteilen, siehe Bild 3:

- NIOS II Softcore-CPU mit Peripherie,
- OFDM-Sender im Basisband (TopOfdm) ,
- digitale Aufwärtsmischung (UpConversion) und
- die Synchronisation.

Die Aufgabe der NIOS II CPU besteht hauptsächlich im Empfang der Sendedaten und der Konfiguration der Sender. Neben der CPU werden noch weitere Komponenten benötigt. Diese können mit dem *SOPC-Builder* von Altera konfiguriert und durch ein Bussystem verbunden werden. Die benötigte Peripherie umfasst

- DDR2 Speicher-Controller,
- Triple-Speed Ethernet-MAC (TSE-MAC),
- Scatter-Gather DMA Controller (SGDMA),
- Timer, Ein-/Ausgabeports (PIOs),
- Ansteuerung für ein Textdisplay und Grafikdisplay sowie
- Phase Locked Loops (PLLs) zur Takterzeugung.

Synchronisations-Core	Text-Display	Grafik-Display
TopOfdm	NIOS II CPU	SGDMA Rx + Tx
UpConversion		Descriptor Memory
PLLs	Clock Crossing Bridge	TSE-MAC
Timer	DDR-2 Speichercontroller	PIOs

Bild 3: Übersicht des SOC mit den darin enthaltenen Komponenten.

Der DDR2 Speicher-Controller ist für den Zugriff der CPU auf den externen DDR2 Speicherbaustein erforderlich. Die Firmware und die Daten sind darin abgelegt. Damit das gesamte System nicht mit der hohen Taktrate des Speichers arbeiten muss, wird dieser Controller über eine *Clock Crossing Bridge* zur Taktentkopplung mit der CPU verbunden. Die Synthese, Platzierung und das Routing am FPGA werden somit vereinfacht. Der TSE-MAC übernimmt die Sicherungsschicht des Ethernet-Protokolls und benötigt zusätzlich spezielle Descriptor-Speicher (im externen DDR2 Speicher) sowie entsprechende SGDMA-Controller. Diese sind für den Sendepfad separat zu implementieren. Wird z.B. ein Datenpaket über die Netzwerkschnittstelle empfangen, wird dieses automatisch vom SGDMA-Controller in den Descriptor-Speicher transferiert und Statusbits werden gesetzt. Beim Senden von Daten läuft dieser Vorgang entsprechend umgekehrt ab. Für die Ausführung des TCP/IP Protokoll-Stacks ist die Verwendung eines Echtzeit-Betriebssystems erforderlich, welches für den Scheduler einen Timer

benötigt.

Für die Anzeige wichtiger Statusinformationen und der aktuellen Senderkonfiguration werden Komponenten zur Ansteuerung der Displays eingebunden. Für einfache Textdisplays bietet Altera bereits passende Hardware- und Softwarekomponenten zur direkten Verwendung an. Zur Ansteuerung des Grafikdisplays wird ein konventioneller Ein-/Ausgabeport verwendet, dessen Pins über die Firmware angesteuert werden. Sämtliche Routinen für das Grafikdisplay sind somit in Software implementiert.

Die Konfiguration der Sender wird in Register abgespeichert, die sich in der Komponente *TopOfdm* befinden. Damit die NIOS II CPU auf diese Register zugreifen kann, müssen diese über eine Schnittstelle mit der CPU verbunden werden. Hierfür wird das *Avalon-MM* Bus-Interface verwendet, welches einen Adress-Datenbus mit Steuerleitungen für den Zugriff darstellt. Die Register werden somit direkt in den Speicherbereich der NIOS II CPU abgebildet. Über Routinen im Hardware-Abstraktions-Layer (HAL) kann von der Firmware aus auf diese Register wie Variablen zugegriffen werden. Die Komponente *TopOfdm* enthält des Weiteren noch, siehe Bild 4:

- eine Komponente zum Lesen und Schreiben der OFDM-Rahmen vom bzw. in das externe SRAM (*LoadSymbol*),
- einen Zwischenspeicher je Sender für zwei OFDM-Symbole (*Symbol Buffer*),
- den eigentlichen OFDM-Core (1 je Sender) und
- eine Komponente zur Steuerung des OFDM-Cores (1 je Sender).

Die Komponente *LoadSymbol* ist ebenfalls über das Bus-Interface mit der CPU verbunden, wodurch das Speichern der OFDM-Rahmen durch die CPU ermöglicht wird. Bei aktivierten Sendern ist ein kontinuierliches Lesen der OFDM-Symbole, bzw. des OFDM-Rahmens erforderlich. Ein direkter und exklusiver Zugriff auf einen externen Speicher ist somit erwünscht. Das Stratix III DSP Board verfügt über einen separaten synchronen RAM-Baustein (SRAM) mit ausreichender Größe für zwei OFDM-Rahmen. Für einen SRAM-Baustein ist kein spezieller Speicher-Controller erforderlich, sodass direkt darauf zugegriffen werden kann. Da nur ein SRAM-Chip zur Verfügung steht, dieser aber Daten für zwei Sender beinhaltet, muss dieser entsprechend schneller getaktet werden. Für Real- und Imaginärteil eines Subträgers werden je 8 Bits reserviert, in Summe also 16 Bits pro Subträger. Da das SRAM eine Datenbreite von 32 Bits aufweist, können mit einem Zugriff entsprechend Daten für zwei Subträger geladen werden, wodurch sich die erforderliche Taktrate wieder reduziert. Die Zwischenspeicher (*Symbol Buffer*) befinden sich im FPGA-internen Speicher und bieten Platz für zwei OFDM-Symbole je Sender. Im Doppelbuffer-Prinzip werden somit abwechselnd in einen Speicher die Daten vom externen SRAM geschrieben und im anderen Speicher bereits geladene Daten dem OFDM-Core zugeführt. Für das Lesen der Symbole aus diesem Zwischenspeicher und den Transfer zum OFDM-Core ist die Komponente *OFDM-Control* zuständig. Des Weiteren lädt diese Komponente den OFDM-Core mit den Werten aus den Konfigurationsregistern und generiert die benötigten Steuersignale für den enthaltenen IFFT-Block. Dieser IFFT-Core mit

variabler Transformationslänge verwendet das von Altera eingesetzte Avalon-Streaming-Interface (Avalon-ST). Der OFDM-Core enthält neben dem IFFT-Block auch einen Block zur Einfügung des Cyclic-Prefix. Die Basis dieser Implementierung beruht auf einer Application-Note von Altera mit geringfügigen Änderungen für unsere Anforderungen [7]. In diesem Beispiel werden das Bit-Reversal der IFFT und das Einfügen des Cyclic-Prefix in einem Schritt durchgeführt. FPGA-interner Speicher wird somit eingespart und steht anderen Komponenten zur Verfügung. Am Ausgang dieses Cores liegt nun das komplexe Basisbandsignal im Zeitbereich vor. Dieses wird anschließend mit einem digitalen Up-Converter (DUC) auf eine Zwischenfrequenz hochgemischt, siehe Kap. 4.

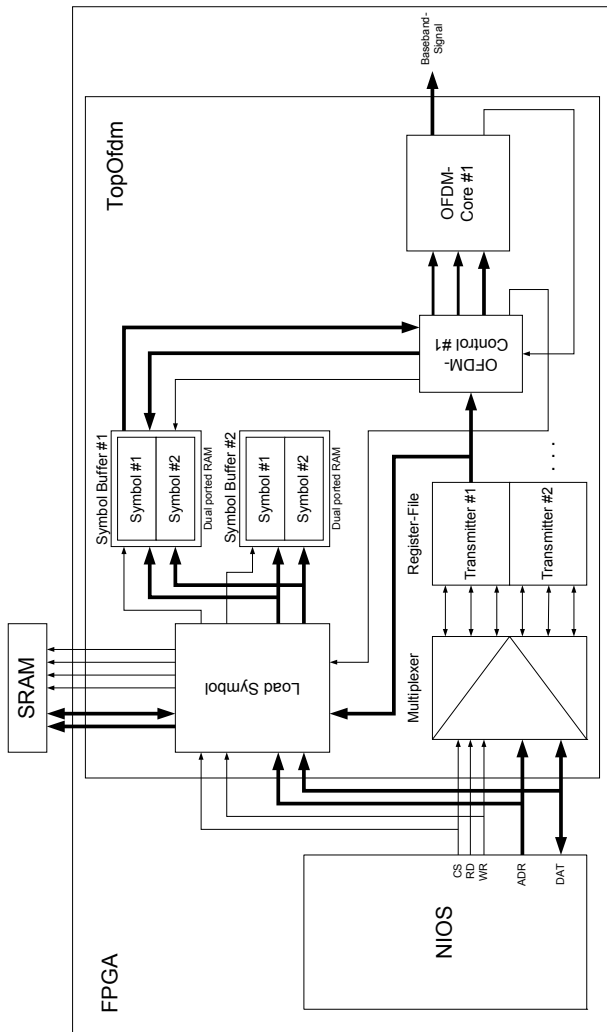


Bild 4: Details der Komponente *TopOfdm* mit eingezeichnetem Datenpfad eines Senders.

Nach dem DUC wird das ZF-Signal an den DAC übertragen. Neben den Daten muss der DAC außerdem mit einem 240 MHz Taktsignal versorgt werden, mit dessen steigender Flanke die Daten in den DAC übernommen werden. Dabei ist zu beachten, dass Setup- und Holdzeiten an den digitalen DAC-Eingängen eingehalten werden. Auf Grund der geometrischen Abmessungen des Stratix III DSP Boards, der ADC/DAC-Zusatzplatine und der auf diesen Boards eingesetzten Taktbuffer ergeben sich erhebliche Leitungslängen und Signallaufzeiten. Die Signallaufzeiten der Daten sind in

Relation zum Taktsignal soweit verschoben, dass der FPGA-intern verwendete 240 MHz Takt nicht verwendet werden kann. Permanente Setup- und Holdzeit-Verletzungen sind die Folge. Zur Kompensation dieser Effekte wird eine zusätzliche interne PLL eingesetzt, welche zwei zusätzliche, unterschiedlich phasenverschobene Referenztaktsignale erzeugt. Erst diese beiden Referenztaktsignale werden den beiden DAC-Kanälen zugeführt. Nachdem sich die Laufzeiten im FPGA auch nach einer neuerlichen Synthese des Designs leicht verändern können, ist eine feste Einstellung dieser Phase zur Synthesezeit schwierig. Die PLLs des Stratix III bieten allerdings die Möglichkeit einer dynamischen Rekonfiguration zur Laufzeit. Die hierfür benötigten Steuerleitungen sind über ein PIO mit der NIOS II CPU verbunden. Die Phase der beiden Referenztaktsignale kann somit über die Firmware und damit auch vom PC aus nachträglich beeinflusst werden. Mittels Oszilloskop kann dann die zeitliche Beziehung zwischen dem Taktsignal und den Daten geprüft und gegebenenfalls sofort im Betrieb angepasst werden. Somit ist gewährleistet, dass beide DAC-Kanäle die Daten korrekt übernehmen. Ist das Design endgültig fixiert, können diese Phasen auch einmalig ausgemessen und permanent als Standardwert in der Firmware gesetzt werden.

#### 4. DIGITALER UP-CONVERTER

Eine digital/analog Wandlung des komplexen Basisbandsignals  $s_{BB}(t) = s_I(t) + js_Q(t)$  mit 20 MHz Bandbreite erfordert zwei DAC-Kanäle pro Sender und sehr steilflankige Analogfilter zur Unterdrückung der Wiederholerspektren. Die eingesetzten DAC-Kanäle erreichen jedoch eine Abtastrate von 250 MSamples/s welche entsprechend ausgenutzt werden kann. Bei z.B. 240 MHz Abtastrate benötigt man nur noch ein analoges Tiefpassfilter mit einem Übergangsbereich von 10 MHz bis 110 MHz. Des Weiteren werden Dämpfungseffekte des DACs im Nutzsband reduziert. Der DAC-Ausgang wirkt als Rechteckimpulsformer, was einer Multiplikation der Impulsantwort mit dem digitalen Signal im Frequenzbereich entspricht. Bei einer Wandlung mit der Nyquist-Rate sind die Einfügedämpfungen entsprechend groß. Eine Vorverzerrungsstufe (Equalizer) müsste diesen Effekt kompensieren [8].

Da im System nur ein DAC-Kanal pro Sender zur Verfügung steht, muss das komplexe Basisbandsignal zuerst in ein reellwertiges Bandpasssignal übergeführt werden. Dies erfolgt mit einem digitalen Quadraturmodulator (Direct Digital Upconversion) auf eine variable Zwischenfrequenz  $f_0$ . Das Bandpasssignal ergibt sich zu  $s(t) = \text{Re}\{s_{BB}(t) * e^{j2\pi f_0 t}\}$ , siehe Bild 5.

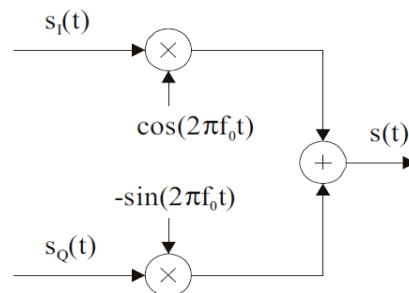


Bild 5: Quadraturmodulator.

Die Zwischenfrequenz wird dabei digital über einen Numerically Controlled Oscillator (NCO) erzeugt. Da der NCO ebenfalls mit 240 MHz getaktet wird, wurde eine Zwischenfrequenz von  $\frac{1}{4} \cdot 240 \text{ MHz} = 60 \text{ MHz}$  gewählt. Dieses Verhältnis hat den Vorteil, dass das komplexe Basisbandsignal nur mit  $0, \pm 1$  und  $\pm j$  multipliziert werden muss, siehe Bild 6. Des Weiteren treten durch dieses ganzzahlige Verhältnis keine Außerband-Artefakte durch Rundungsfehler in der Mischerstufe bzw. im NCO auf.

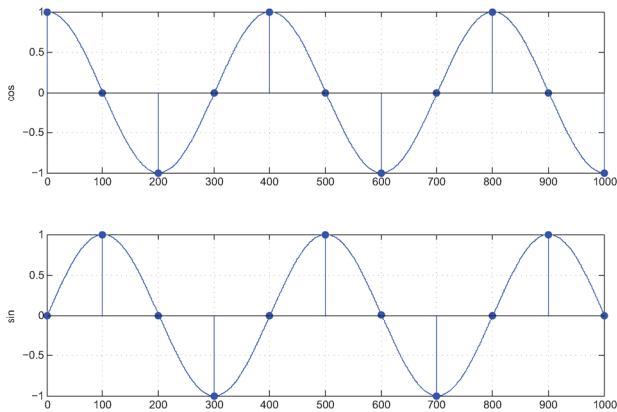


Bild 6: Signalverlauf am NCO Ausgang bei 240 MHz Systemfrequenz und 60 MHz Signalfrequenz.

Der NCO wäre bei dieser Konfiguration nicht erforderlich, aus Gründen der Flexibilität wird dieser jedoch beibehalten. Die Aufwärtstastung von 20 MHz auf 240 MHz erfolgt in Hardware durch einfügen von Nullen (Zero-Padding) und anschließender Tiefpassfilterung (Interpolation). Für eine effiziente Implementierung bietet es sich an, den Interpolationsfaktor von  $240\text{MHz}/20\text{MHz} = 12$  nicht auf einmal sondern schrittweise zu realisieren [9; 10]. Die Anforderungen an die Interpolationsfilter sind somit geringer und der Hardwareaufwand wird reduziert. Das Bild 7 zeigt die Gesamtstruktur der Aufwärtstastung und des Mixers.

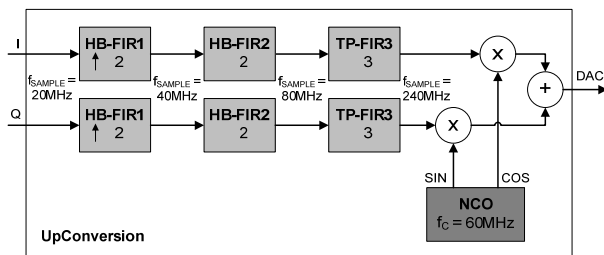


Bild 7: Aufwärtstaster mit Interpolationsfilter und digitalem Mischer.

Besonders geeignet sind Interpolationen um den Faktor zwei, welche durch Halbband-FIR-Filter effizient realisiert werden können. Bei diesen FIR-Filtern, bei welchen jeder zweite Koeffizient 0 ist, können durch geeignete Polyphasenstrukturen Multiplikationen (Koeffizienten) eingespart werden. Die Interpolation wird in zwei Halbband-FIR-Filter mit einem Interpolationsfaktor von je zwei und ein gewöhnliches FIR-Filter mit Faktor 3 aufgeteilt. Das erste Filter benötigt die meisten Koeffizienten da es entsprechend steilflankig sein muss, siehe Bild 8.

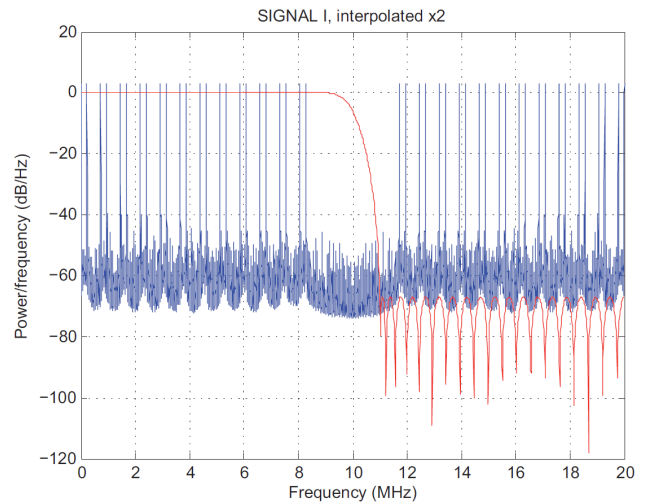


Bild 8: Spezifikation des ersten Halbband-FIR-Filters mit eingezeichnetem OFDM-Spektrum.

Da bei OFDM-basierten Systemen ohnehin stets ein Guard-Band spezifiziert wird, orientiert sich die Komplexität des ersten Filters an Spezifikationen von z.B. 3GPP-LTE oder DVB-T. Im Guard-Band werden die entsprechenden Subträger vor der IFFT zu Null gesetzt um einen Übergangsbereich ohne Nutzsignalleistung zu schaffen [11]. In diesem Design wurde ein Guard-Band von 1 MHz definiert (3 dB Cut-off bei 9 MHz). Bei entsprechend steilflankigeren Filtern kann die Anzahl der genutzten Unterträger noch weiter erhöht werden. Das zweite und dritte Filter kann, bei genauer Betrachtung der Spektren, einen flacheren Übergangsbereich aufweisen. Die Anzahl der benötigten Koeffizienten zur Realisierung wird somit minimiert, siehe Bild 9 und Bild 10.

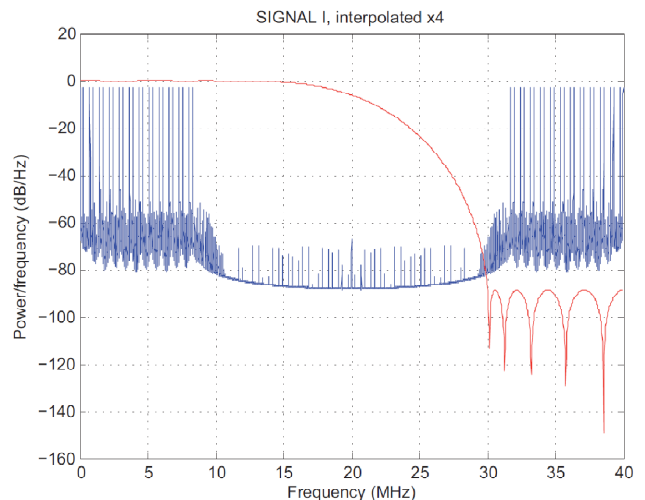


Bild 9: Spezifikation des zweiten Halbband-FIR-Filters mit eingezeichnetem OFDM-Spektrum.



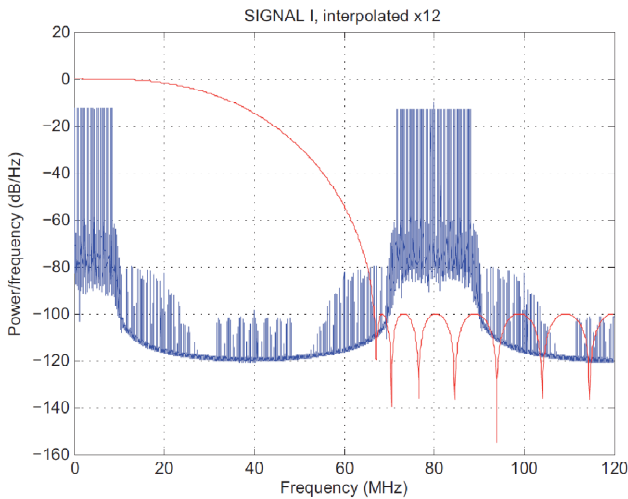


Bild 10: Spezifikation der letzten FIR Filterstufe mit eingezeichnetem OFDM-Spektrum.

Die Tab 1 fasst die Filterspezifikationen und die benötigten Koeffizienten zusammen.

Parameter	Stufe 1	Stufe 2	Stufe 3
Passband-Ripple	0.1 dB	0.1 dB	0.1 dB
3 dB Cut-off	9 MHz	10 MHz	30 MHz
Sperrbereichs-Dämpfung	> 70 dB	> 70 dB	> 70 dB
Koeffizienten	74	18	16

Tab 1: Zusammenfassung der Filterspezifikationen.

Würde man im Vergleich dazu ein einziges FIR- Interpolationsfilter auf der hohen Abtastrate entwerfen, ergibt sich eine Gesamtkoeffizientenzahl von 363, siehe Tab 2.

Parameter	Stufe 20 MHz – 240 MHz
Passband-Ripple	0.1 dB
3 dB Cut-off	9 MHz
Sperrfrequenz	11 MHz
Sperrbereichs-Dämpfung	> 70 dB
Koeffizienten	363

Tab 2: Spezifikation eines FIR-Gesamtfilters äquivalent zur Gesamtstruktur in Bild 3.

## 5. SYNCHRONISATION

Die Genauigkeit der Positionsbestimmung nach dem TDoA-Verfahren hängt unter anderem von folgenden Faktoren ab:

- Der Nutzsignal-Bandbreite,
- dem Signal- zu Rauschleistungsverhältnis,
- der geometrischen Anordnung der beteiligten Basisstationen und
- der zeitlich exakten Abstrahlung der Signale von den Basisstationen.

Vor allem der letzte Punkt ist besonders kritisch. Die beteiligten Stationen sollten zeitgleich ihre Signale abstrahlen, also zeit- und phasensynchron. Diese Anforderung lässt sich vereinfachen, indem man z.B. die unterschiedlichen Abstrahlzeitpunkte einmalig vermisst. Der mobile Empfänger kann diese festen Verzögerungen bei der Auswertung der Ergebnisse berücksichtigen. Dann müssen diese jedoch stets konstant sein und dürfen sich bei einem Neustart oder einer Neukonfiguration des Systems nicht ändern. Diese Konstanz lässt sich im Betrieb jedoch nur schwer erreichen. Des Weiteren ist zu beachten, dass die räumlich verteilten FPGA-Boards hinsichtlich folgender Punkte Unterschiede aufweisen:

- Frequenz, Frequenz- und Phasenstabilität (Jitter) des Taktsignals,
- Fertigungstoleranzen der Chips, Leiterplatten und Verbindungskabel,
- Temperaturschwankungen der Boards sowie Spannungsversorgungsschwankungen,
- Länge und Umgebungssituation der Verbindungen zwischen den FPGA-Boards.

Bis auf den ersten Punkt führen alle Punkte zu Änderungen der Laufzeit und der Ausbreitungsgeschwindigkeit im jeweiligen Medium. In der Digitaltechnik spielt die Einsynchronisation externer Signale in eine bestimmte Taktdomäne eines Chips stets eine große Rolle. Werden Setup- oder Hold-Zeiten verletzt, wird der Chip metastabil und eine korrekte Funktion kann nicht mehr gewährleistet werden. In diesem Fall hilft auch eine externe hochgenaue Taktgeberquelle nicht, da die Einsynchronisierung fehlschlagen kann. Viele einfache niederrangige Synchronisationsmethoden auf Bit-Ebene bestehen aus Kaskaden von Flip-Flops oder speziellen asynchronen Speichern (Dual-Ported RAMs). Die Wahrscheinlichkeit eines metastabilen Zustandes wird somit auf ein Minimum reduziert. Es lässt sich jedoch nicht feststellen, ob das Signal nun 1 oder 2 oder mehr Takte zur Stabilisation durch die Synchronisationsstufen benötigte.

Die weiteren Anforderungen an das System sind somit die Vermeidung der konventionellen Einsynchronisation und der Laufzeitausgleich auf den Verbindungskabeln zwischen den FPGA-Boards. Eine automatische Nachjustierung zum Ausgleich von Effekten durch Temperatur- und Spannungsschwankungen sollte ebenfalls stattfinden. Für aktuelle, serielle Hochgeschwindigkeitsverbindungen (> 200 Mbit/s) gibt es jedoch Technologien zur dynamischen Phasenanpassung im Empfangszweig [12; 13; 14]. Aktuelle Speichertechnologien erfordern ebenfalls spezielle Anpassungstechniken zum Laufzeitausgleich auf Leiterplatten (PCBs) [15].

Die entwickelte Lösung verwendet bereits etablierte Hochgeschwindigkeitsverbindungen, in Kombination mit einem Master/Slave System und einer Laufzeitmessung zwischen den FPGA-Boards. Damit beide FPGA-Boards mit demselben Hardware-Design geladen werden können, ist der Synchronisations-Core generisch und für beide ident. Für die NIOS II CPU wird auch eine einheitliche API zur Bedienung der Links bereitgestellt.

Der Synchronisations-Link arbeitet gegenwärtig mit konventionellen Koaxialkabeln und dem LVDS-Signalstandard [16]. Zur Überbrückung größerer Entfernungen und zur Vermeidung von zu großen Kabeldämpfungen können auch optische Verbindungen eingesetzt werden.

## 6. BENUTZEROBERFLÄCHE

Zur Kommunikation wurde ein Protokoll für die Konfiguration der OFDM-Cores und die Übertragung der Sendesymbole implementiert. Bei der Serialisierung der Daten ist auf die Ausrichtung der gesendeten 32 bit Werte zu achten (Endianess), da die PC-Architekturen und die NIOS II Architektur Unterschiede aufweisen. Die grafische Benutzeroberfläche (GUI) wird in Matlab erzeugt um möglichst einfach zwischen verschiedenen PC-Betriebssystemen wechseln zu können. Mit dem, in Matlab eingebauten, Assistenten GUIDE können entsprechende Elemente auf dem GUI platziert werden. Sämtliche weitere Funktionalitäten und das Verhalten der einzelnen Elemente werden mit gesonderten Matlab Scripts behandelt.

Die Oberfläche ist in 5 Bereiche gegliedert, siehe Bild 11:

- 1) Darstellung des Frameaufbaus,
- 2) Parameter der einzelnen Frames,

- 3) Ausgabefenster für Statusmeldungen (Konsole),
- 4) FPGA-Board Befehle und
- 5) Generierung und Befüllung der Rahmen.

Zu Beginn müssen die Parameter für jeden einzelnen Sender eingestellt werden, z.B. Anzahl der Subträger, Cyclic Prefix, etc. Danach werden die Rahmen mit benutzerdefinierten Daten gefüllt. Zur Erzeugung dieser Daten kann eine externe, selbst geschriebene Matlab-Funktion übergeben werden. Nach der Eingabe der IP-Adressen der Boards wird eine Verbindung aufgebaut. Gleichzeitig mit dem Verbindungsaufbau wird auch die Synchronisation zwischen den Boards überprüft. Soll zu Testzwecken nur ein Board aktiviert sein, kann die Synchronisation im Betrieb abgeschaltet werden. Des Weiteren kann der Synchronisations-Core aufgefordert werden, sich erneut zu synchronisieren, falls die Verbindung zwischen den FPGA-Boards unterbrochen wurde. Nach diesem Verbindungsaufbau können die beiden Boards konfiguriert und gestartet werden. Damit sämtliche Einstellungen und Daten der Rahmen nicht erneut eingegeben werden müssen, erfolgt eine Speicherung in einer gesonderten Datei. Die gespeicherten Einstellungen können später wieder geladen werden.

Die Erzeugung der Sendedaten für die einzelnen Frames wird über eine externe, einstellbare Matlab-Funktion erreicht. Informationen bezüglich der Frame-Längen etc. sind darin abrufbar um die Arrays korrekt zu füllen. Beim Erzeugen ist darauf zu achten, dass die gemappten Datensymbole (BPSK, QPSK, 16-QAM, 64-QAM, etc.) anschließend auf 8 Bit Werte je I/Q-Komponente quantisiert werden.

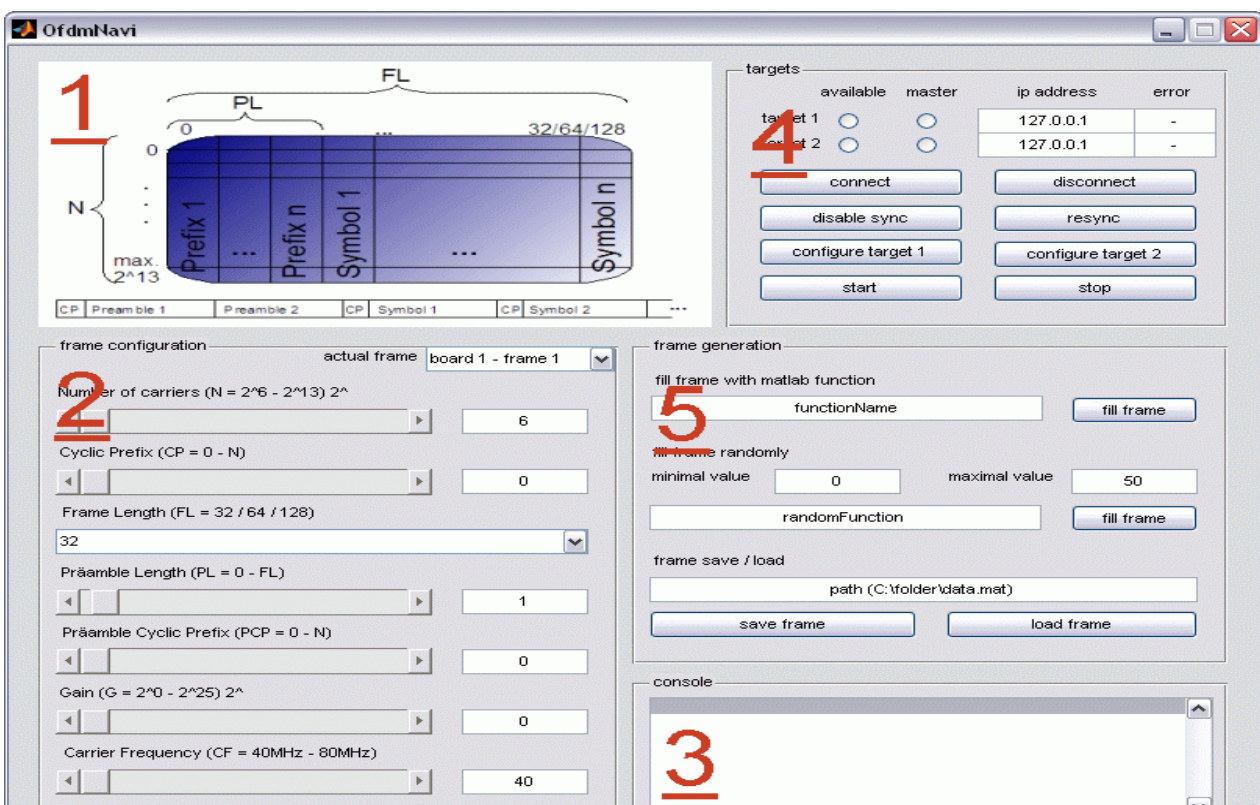


Bild 11: Darstellung der grafischen Benutzeroberfläche.

## 7. FIRMWARE

Die Firmware stellt das Bindeglied zwischen der Benutzerschnittstelle am PC, dem OFDM-Sender und dem Synchronisationsmodul dar. So werden Befehle und Daten vom PC empfangen, verarbeitet und Statusmeldungen an den PC zurückgesendet. Eine wichtige Funktion besteht darin, die Sendedaten vom PC entgegen zu nehmen, die Datenpakete zusammen zu setzen und in den Zwischenspeicher der OFDM-Sender zu schreiben.

Für die Entwicklung der Software auf dem NIOS II Softcore stellt Altera die *NIOS II Embedded Design Suite* Toolchain bereit. Diese beinhaltet unter anderem Compiler, Linker, Debugger sowie eine Umgebung auf Basis der *C/C++ Development Tools* (CDT) für die weit verbreitete Eclipse IDE. Zusammen mit der Toolchain werden eine Implementierung der C-Laufzeitbibliotheken und Beispielprojekte ausgeliefert, wie z.B. ein Ethernet Server auf Basis von Sockets. Dieses Beispielprojekt wurde als Grundlage für die weiteren Entwicklungen herangezogen und schrittweise erweitert.

In diesem Projekt wird ein Echtzeitbetriebssystem (RTOS) der Firma Micrium eingesetzt. Dieses RTOS namens  $\mu\text{C}/\text{OSII}^{\text{TM}}$  wurde bereits speziell für den NIOS II portiert. Der Einsatz eines RTOS ermöglicht eine strukturierte Verwaltung der Abläufe durch nebenläufige Prozesse und beschleunigt so den Entwicklungsprozess. Zusätzlich bietet es eine gute Grundlage für zukünftige Erweiterungen. Die Firmware selbst ist in mehrere Module aufgeteilt, welche jeweils einem Aufgabenbereich entsprechen: Überwachung der Synchronisation, Verwaltung der Benutzerschnittstellen am FPGA-Board, Kommunikationsprotokoll und Konfiguration der PLL-Phasen. Diese Module bieten zudem jeweils die Möglichkeit einer Ausgabe von Debuginformationen über eine sichere Standardausgabe (threadsafe).

Der Status der Synchronisation kann vom PC abgefragt werden. Die Firmware überprüft dabei die Hardware-Register des Synchronisationsmoduls und sendet entsprechende Antworten an den PC zurück. Zusätzlich kann die Synchronisation deaktiviert bzw. neu gestartet werden. IP- bzw. MAC-Adresse sowie die Master-/Slave-Einstellungen für die Synchronisation werden vom Benutzer direkt an den FPGA-Boards mit DIP-Schaltern vorgenommen. Diese Zustände werden beim Start der Firmware automatisch eingelesen und ausgewertet. Die dabei eingestellte IP-Adresse sowie der Master-/Slave-Status werden direkt auf einem LCD ausgegeben. LEDs zeigen den gespeicherten Zustand der DIP-Schalter an, damit eine Änderung der Schalterstellung während des Betriebs nicht zu falschen Annahmen führt. Sämtliche, für die OFDM-Sender relevanten Konfigurationsinformationen werden auf einem Grafik-Display visualisiert, nachdem die Konfiguration abgeschlossen wurde.

Die Kommunikation zwischen dem PC und den FPGA-Boards findet über Ethernet statt, wobei ein Server-Client Prinzip zum Einsatz kommt. Die beiden FPGA-Boards fungieren dabei jeweils als Server und der PC als Client. Die Firmware wartet auf Anfragen des Clients bzw. auf Anfragen für einen Verbindungsaufbau und sendet entsprechende Antworten zurück. Für die Kommunikation selbst werden einfache Datenpakete verwendet. Diese bestehen aus einem Headerbereich und einem Datenbereich. Der Headerbereich beinhaltet Kontrollinformationen wie z.B. eine Kommando-ID, Längen-

angaben des Nutzdatenbereichs etc. Für die Übertragung selbst wird ein kommerzieller TCP/IP-Stack der Firma *Interniche Technologies Inc.* eingesetzt, welcher für den Softcore-Prozessor und das RTOS portiert wurde. Diese Portierung ist im Softwarepaket von Altera bereits enthalten. Der Entwickler kann mit diesem Stack Berkeley-konforme Sockets – die de facto Standard Socket API – für die Zugriffe auf diese Ethernet-Kommunikationschnittstelle verwenden.

## 8. ERGEBNISSE

Durch Simulationen des OFDM-Cores und spektrale Messungen des Sendesignals kann die Funktion des Gesamtdesigns überprüft werden. Das Bild 12 zeigt beispielhaft das Spektrum eines Ausgangssignals bei 8192 Subträgern, wobei jedoch nur wenige Subträger selbst aktiviert sind.

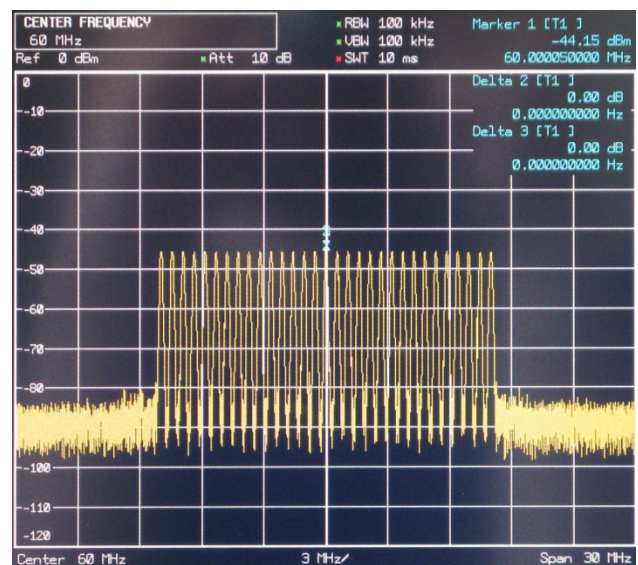


Bild 12: OFDM Spektrum auf ZF = 60 MHz und wenigen aktiven Subträgern.

In Bild 13 sind das Taktsignal eines DAC-Kanals und ein Datenbitsignal abgebildet. Das obere Signal zeigt den Takt und das untere das Datum. In dieser Darstellung ist ersichtlich, dass die steigende Flanke des Taktsignals nahezu exakt in der Mitte eines Datenbits liegt d.h. die Phase wurde korrekt über das Benutzer-Interface eingestellt.

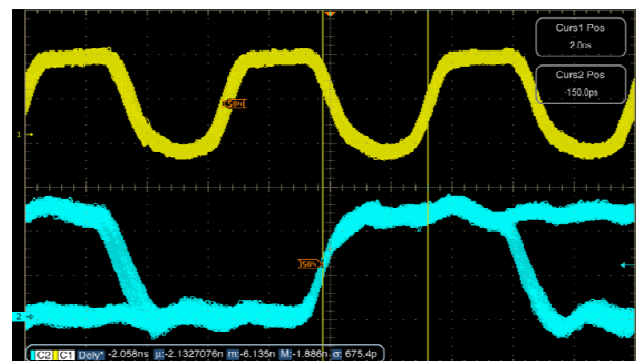


Bild 13: Mehrfache Aufnahme des Taktsignals (oben) und eines Datenbits (unten) eines DAC-Kanals.



Die Einstellungen für den Synchronisations-Link wie z.B. Vorverzerrung oder Signalpegelhöhe werden in einem eigenen Testaufbau überprüft. Als tolerierbare Bitfehler-rate wird jene von aktuellen Hochgeschwindigkeits-standards wie z.B. PCI-Express verwendet, welche  $10^{-12}$  bei einem Confidence Level von ca. 99.5% beträgt [17; 18].

Die Ressourcen der beiden FPGAs wurden nicht zur Gänze ausgeschöpft. Es werden nur ca. 50% der internen Speicherblöcke und ca. 75% der ALUTs benötigt, was Spielraum für zukünftige Erweiterungen lässt. Durch die Phasenanpassung an den einzelnen DAC-Kanälen der 4 Sender tritt eine zusätzliche, unterschiedliche Laufzeit der Sendesignale auf. Da jedoch die Phaseneinstellungen und die PLL-Einstellungen bekannt sind, können diese fixen Laufzeiten im mobilen Empfänger berücksichtigt werden.

Double Data Rate LVDS Links. Universitat de València- Estudi General : s.n., 2004. Doktorarbeit.

[14]. **Xilinx**. Virtex-5 FPGA RocketIO GTX Transceiver. 2009. User Guide UG198.

[15]. *Calibration Techniques for High-Bandwidth Source Synchronous Interfaces*. **Roge, Manoj, et al.** s.l. : DesignCon, 2007.

[16]. **National, Semiconductor**. *LVDS Owner's Manual, Including High-Speed CML and Signal Conditioning*. 2008.

[17]. **Maxim Integrated Products**. Statistical confidence levels for estimating error probability. *Lightwave Magazine*. 2000.

[18]. **Martwick, A.** *PCI Express Architecture, PCI Express Jitter and BER*. s.l. : PCI-SIG, 2005. Bd. Revision 1.0.

## 9. LITERATURVERZEICHNIS

[1]. *Positioning in Wireless Communications Systems - Introduction and Overview*. **Sand, Stephan, Mensing, Christian und Damman, Armin**. Espoo, Finnland : 18th Wireless World Research Forum (WWRF) Meeting, 2007.

[2]. *The WHERE Project – Combining Wireless Communications and Navigation*. **Raulefs, Ronald, Plass, Simon und Mensing, Christian**. Ottawa, Canada : 20th World Wireless Research Forum Meeting (WWRF), 2008.

[3]. *Providing Location Service for Mobile WiMAX*. **Jiao, Wenhua, et al.** Beijing : IEEE International Conference on Communications, ICC, 2008.

[4]. **Yu, Kegen, Sharp, Ian und Guo, Y. Jay**. *Ground-Based Wireless Positioning*. s.l. : John Wiley & Sons Ltd., 2009.

[5]. *Synchronization Algorithms for Positioning with OFDM Communications Signals*. **Mensing, Christian, Plass, Simon und Dammann, Armin**. Hannover, Germany : Workshop on Positioning, Navigation and Communication (WPNC), 2007.

[6]. *Positioning Algorithms for Cellular Networks Using TDoA*. **Mensing, Christian und Plass, Simon**. Toulouse, France : IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), 2005.

[7]. **Altera**. *Implementing OFDM Modulation for Wireless Communications*. 2008. Application Note. 503.

[8]. **Zölzer, Udo**. *Digitale Audiosignalverarbeitung*. s.l. : Vieweg+Teubner, 2005.

[9]. **Fliege, Norbert**. *Multiraten-Signalverarbeitung*. s.l. : Teubner, 1993.

[10]. *A digital IF-modulator for video and associated audio signals*. **Wenzel, D. und Speidel, J.** Vancouver : IEEE International Conference on Communication, ICC, 1999.

[11]. **Sesia, Stefania, Toufik, Issam und Baker, Matthew**. *LTE - The UMTS Long Term Evolution, From Theory to Practice*. s.l. : John Wiley & Sons Ltd., 2009.

[12]. **Corporation, Altera**. *High-Speed Differential I/O Interfaces and DPA in Stratix III Devices*. 2008. Stratix III Device Handbook.

[13]. **Torrallba Collados, Gloria Mariá**. *Design of a Deskewing and Fault Tolerance Switch for High Speed*